

INICIAR SESIÓN

NUESTROS PLANES

TODOS LOS  
CURSOS

FORMACIONES

CURSOS

PARA  
EMPRESAS

ARTÍCULOS DE TECNOLOGÍA &gt; FRONT END

# Manipulación de array con map, filter y reduce



Felipe Nascimento

07/10/2021

Aprende a manipular matrices de alguna manera **funcional** en Javascript, usando los métodos de map, reduce y filter.

En la empresa donde trabajo, iniciamos un [Codigng Dojo](#), que es básicamente una reunión con programadores para resolver desafíos y mejorar sus habilidades algorítmicas.

El desafío de esta semana fue el siguiente, dado el array empresas:

```
const empresas = [
```

```
{ nombre:'Samsung',valorDeMercado: 50, CEO: 'Kim Hyun Suk', añoDeCreacion: 1938},  
{ nombre: 'Microsoft',valorDeMercado: 415, CEO: 'Satya Nadella', añoDeCreacion: 1975 },  
{ nombre: 'Intel',valorDeMercado: 117, CEO:'Brian Krzanich', añoDeCreacion: 1968},  
{ nombre: 'Facebook',valorDeMercado: 383, CEO:'Mark Zuckerberg', añoDeCreacion: 2004},  
{ nombre: 'Spotify',valorDeMercado: 30, CEO:'Daniel Ek', añoDeCreacion: 2006 },  
{nombre: 'Apple', valorDeMercado: 845, CEO: 'Tim Cook', añoDeCreacion: 1976}];
```

Tengo que mostrar la siguiente información:

- Empresas creadas después de los años 2000.
- El nombre de cada empresa y su CEO.
- El valor de todas las empresas sumadas.

Lo primero que debemos mostrar son las empresas creadas después de los años 2000. Lo que podemos hacer en este caso es crear una forma de filtrar el array para poder mostrar esta información.

## Filter

Por suerte ya existe un método para hacer este tipo de filtro y ¿adivina su nombre? Eso mismo [filter](#). Pasará por todo el array y creará uno nuevo con todos los elementos que pasaron en la prueba implementada, en nuestro caso, cuando el año de creación sea mayor que 2000.

Como tenemos objetos dentro del array, necesitamos acceder a sus propiedades para hacer el filtro. Para hacer esto, simplemente llame al nombre del array seguido de la notación de punto y luego el nombre de la propiedad.

Entonces para acceder a la propiedad añoDeCreacion hacemos:

```
empresas.añoDeCreacion
```

Ahora simplemente crea la condición para mostrar el añoDeCreacion mayor que 2000, dentro del filter:

```
const añoDeCreacion = empresas.filter(empresa => (empresa.añoDeCreacion > 2000)); console.log( añoDeCreacion );
```

Como resultado tenemos:

```
[ { nombre: 'Facebook', valorDeMercado: 383, CEO: 'Mark Zuckerberg',  
  añoDeCreacion: 2004 }, { nombre: 'Spotify', valorDeMercado: 30, CEO: 'Daniel  
Ek', añoDeCreacion: 2006 } ]
```

Pudimos realizar la primera tarea, ahora para la segunda tenemos que mostrar los nombres de las empresas y sus CEOs. Sería interesante crear una función que extraiga estos dos valores para nosotros.

## Map

Afortunadamente tenemos este método listo y se llama [map](#), con él es posible revisar todos los elementos del array, realizar alguna transformación en estos elementos y devolver un nuevo array sin modificar la actual.

En nuestro caso, revisaremos el array y dentro de cada objeto tomaremos las propiedades nombre y CEO de cada empresa.

```
const exhibeInformaciones = empresas.map( empresa => `${empresa.nombre} + ' CEO: ' +${empresa.CEO});console.log( exhibeInformaciones )
```

Tenemos como respuesta:

```
[ 'Samsung CEO: Kim Hyun Suk', 'Microsoft CEO: Satya Nadella', 'Intel CEO: Brian Krzanich', 'Facebook CEO: Mark Zuckerberg', 'Spotify CEO: Daniel Ek', 'Apple CEO: Tim Cook' ]
```

Un elemento más de nuestra lista de verificación completa, ahora tenemos que mostrar el valor de todas las empresas sumadas. Sería bueno si ya tuviéramos algo similar a los dos últimos métodos, pero sumar todos los valores.

## Reduce

Y tenemos el método [reduce](#) pasa a través de cada elemento del array haciendo una expresión elegida, y al final devolverá un solo valor. Exactamente lo que necesitamos para hacer la suma del valor de mercado de las empresas.

Recibe dos valores, uno acumulado y otro actual, en nuestro caso el acumulado se sumará al actual. Como en el primer loop de la suma no tenemos ningún valor acumulado, por lo que debemos pasar este valor como segundo parámetro, que en nuestro caso será 0.

Si no hubiéramos pasado un valor acumulado, reduce tomaría el primer elemento del array, tomándolo como el valor acumulado. Nuestro código estaba terminado y se veía así:

```
const total = empresas.reduce((resultado, cantidad) => { return (resultado + cantidad.valorDeMercado ')}), 0);
```

Como resultado tenemos:

```
console.log( total )
```

1840

Con el uso de métodos como map, filter y reduce todos los desafíos propuestos en Coding Dojo fueron posibles con códigos semánticos, sencillos y funcionales.

Si estás interesado en cómo funciona Javascript y cómo puedes usarlo mejor, aquí en Alura tenemos una [formación front-end](#). Ahí, verás cómo utilizar otros métodos de Javascript moderno, conocerás buenas prácticas y mucho más.



### Felipe Nascimento

Desarrollador e instructor en Alura con enfoque en JavaScript

Puedes leer también:

- [Llenando un formulario HTML automáticamente con AJAX](#)
- [HTML, CSS y Javascript, ¿cuáles son las diferencias?](#)
- [Empezando con fetch en Javascript](#)

ARTÍCULOS DE TECNOLOGÍA > FRONT END

**En Alura encontrarás variados cursos sobre Front End.  
¡Comienza ahora!**

**SEMESTRAL**

**US\$49,90**

un solo pago de US\$49,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 6 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

**ANUAL**

**US\$79,90**

un solo pago de US\$79,90

- ✓ 218 cursos
- ✓ Videos y actividades 100% en Español
- ✓ Certificado de participación
- ✓ Estudia las 24 horas, los 7 días de la semana
- ✓ Foro y comunidad exclusiva para resolver tus dudas
- ✓ Acceso a todo el contenido de la plataforma por 12 meses

**¡QUIERO EMPEZAR A ESTUDIAR!**

[Paga en moneda local en los siguientes países](#)

Acceso a todos  
los cursos

Estudia las 24 horas,  
dónde y cuándo quieras

Nuevos cursos  
cada semana

## NAVEGACIÓN

PLANES

INSTRUCTORES

BLOG

POLÍTICA DE PRIVACIDAD

TÉRMINOS DE USO

SOBRE NOSOTROS

PREGUNTAS FRECUENTES

## ¡CONTÁCTANOS!

¡QUIERO ENTRAR EN CONTACTO!

## BLOG

PROGRAMACIÓN

FRONT END

DATA SCIENCE

INNOVACIÓN Y GESTIÓN

DEVOPS

AOVS Sistemas de Informática S.A

CNPJ 05.555.382/0001-33

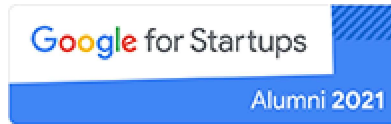
## SÍGUENOS EN NUESTRAS REDES SOCIALES



## ALIADOS



En Alura somos unas de las Scale-Ups seleccionadas por Endeavor, programa de aceleración de las empresas que más crecen en el país.



Fuimos unas de las 7 startups seleccionadas por Google For Startups en participar del programa Growth Academy en 2021

POWERED BY

## CURSOS

### Cursos de Programación

Lógica de Programación | Java

### Cursos de Front End

HTML y CSS | JavaScript | React

### Cursos de Data Science

Data Science | Machine Learning | Excel | Base de Datos | Data Visualization | Estadística

### Cursos de DevOps

Docker | Linux

### Cursos de Innovación y Gestión

Productividad y Calidad de Vida | Transformación Ágil | Marketing Analytics | Liderazgo y Gestión de Equipos | Startups y Emprendimiento